70

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/640,118	08/16/2000	G. Glenn Henry	CNTR:1356	4572

23669 7590 11/16/2005

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 11/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/640,118	HENRY ET AL.
	Examiner	Art Unit
	Aimee J. Li	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 September 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19 and 21-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19 and 21-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 and 21-24 have been considered. Claims 1, 6, 11, 14, and 21 have been amended as per Applicant's request. Claim 20 has been cancelled as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 02 September 2005.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-19 and 21-24 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Each independent claim recites the limitation, taking claim 1 as exemplary, "bypass logic…configured to provide the programmed native instructions directly to execution logic for execution…". After perusing the specification, the Examiner could not find an explanation of how it is possible to directly provide the programmed native instructions directly to execution logic when the system executes multiple languages and a translator is required. Even in Applicant's specification the bypass passes the instructions through a multiplexer before the instruction is provided to the native instruction bus (Applicant's Figure 4), thus the instruction is not directly provided since the connection is interrupted by a multiplexer.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-19 and 21-24 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Hammond et al., U.S. Patent Number 5,638,525 (herein referred to as Hammond) in view of Patterson and Hennessy's <u>Computer Architecture: A Quantitative Approach</u> Second Edition ©1996 (herein referred to as Hennessy).

7. Referring to claim 1, Hammond has taught an apparatus in a microprocessor for executing programmed native instructions that are provided directly to the microprocessor via an external instruction bus (Hammond column 4, lines 16-45 and Figure 1), the apparatus comprising:

a. Instruction translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), configured to retrieve macro instructions provided via the external instruction bus (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and configured to decode each of said macro instructions into associated native instructions for execution by the microprocessor (Hammond column 17, lines 25-48 and 57-63; and Figure 7), wherein said instruction translation logic decodes a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the programmed native instructions (Hammond column 4, line 61 to column 5, line 8;

column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), and

b. Bypass logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), coupled to said instruction translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), configured to disable said instruction translation logic upon detection of said native bypass macro instruction (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and configured to provide the programmed native instructions directly to execution logic for execution, thereby bypassing said instruction translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

8. Hammond has not taught wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction. Hennessy has taught wherein said memory address is explicitly prescribed by contents of an architectural register (Hennessy page 82, paragraph 2), said contents and said architectural register being prescribed by a macro instruction (Hennessy page 82, paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to

jump to the correct address even when the address is not known at the time the program is compiled.

9. Referring to claim 2, Hammond has taught wherein the programmed native instructions are provided from a memory to the external instruction bus (Hammond column 4, lines 16-45 column 17, lines 25-48 and 57-63; Figure 1; and Figure 7).

10. Referring to claim 3, Hammond has taught wherein execution of said native bypass macro instruction causes the microprocessor to transfer program control to the programmed native instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

11. Referring to claim 4, Hammond has taught wherein said bypass logic comprises mode detection logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), configured to detect said native bypass macro instruction within a macro instruction sequence that is provided to said instruction translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), wherein, upon detection of said native bypass macro instruction, said mode detection logic directs said instruction translation logic to cease decoding said macro instruction sequence following decoding of said native bypass macro instruction (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

12. Referring to claim 5, Hammond has taught wherein said unconditional jump native instruction directs the microprocessor to transfer program control to said memory address (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

13. Referring to claim 6, Hammond has taught wherein the microprocessor comprises an x86-compatible microprocessor (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2). Hammond has not taught wherein said architectural register comprises register EAX. Hennessy has taught wherein said architectural register comprises register EAX (Hennessy page 82, paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to jump to the correct address even when the address is not known at the time the program is compiled.

14. Referring to claim 7, Hammond has taught wherein said bypass logic further comprises a native instruction router (Hammond column 17, lines 25-48 and 57-63; and Figure 7), coupled to said mode detection logic configured to receive the programmed native instructions and configured to route the programmed native instructions directly to said execution logic via a native instruction bus (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

15. Referring to claim 8, Hammond has taught wherein, said mode detection logic is also configured to detect a native branch return macro instruction (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), said native branch return macro instruction following the

programmed native instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), wherein, upon detection of said native branch return macro instruction, said mode detection logic directs said instruction translation logic to resume decoding said macro instruction sequence (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

16. Referring to claim 9, Hammond has taught wherein said instruction translation logic decodes said native branch return macro instruction into a native branch return native instruction, and wherein said native branch return native instruction directs the microprocessor to transfer program control to a return address (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

17. Referring to claim 10, Hammond has taught wherein said return address designates a next macro instruction, said next macro instruction being within said macro instruction sequence and following said native branch macro instruction (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

18. Referring to claim 11, Hammond has taught an apparatus, for allowing a micro instruction to be directly provided from an external instruction bus to execution logic within a pipeline microprocessor (Hammond column 4, lines 16-45 and Figure 1), the apparatus comprising:

a. A translator (Hammond column 17, lines 25-48 and 57-63; and Figure 7), for receiving macro instructions from a macro instruction bus (Hammond column 17,

lines 25-48 and 57-63; and Figure 7), and for translating each of said macro instructions into associated micro instructions (Hammond column 17, lines 25-48 and 57-63; and Figure 7), said associated micro instructions being provided to the execution logic via a micro instruction bus (Hammond column 17, lines 25-48 and 57-63; and Figure 7), wherein said translator translates a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the micro instruction (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), and

b. Bypass logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), coupled to said translator, for routing the micro instruction to the execution logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), said bypass logic comprising:

   i. A mode detector, for detecting said native bypass macro instruction, and for directing that said translator cease instruction translation (Hammond column 17, lines 25-48 and 57-63; and Figure 7); and

   ii. Native instruction routing logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), coupled to said mode detector, for receiving said micro instruction from said macro instruction bus (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and for providing said micro instruction directly to the execution logic via said micro instruction bus,

thereby circumventing said translator (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

19. Hammond has not taught wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction. Hennessy has taught wherein said memory address is explicitly prescribed by contents of an architectural register (Hennessy page 82, paragraph 2), said contents and said architectural register being prescribed by a macro instruction (Hennessy page 82, paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to jump to the correct address even when the address is not known at the time the program is compiled.

20. Referring to claim 12, Hammond has taught wherein the external instruction bus typically provides said macro instructions to the microprocessor (Hammond column 4, lines 16-45 column 17, lines 25-48 and 57-63; Figure 1; and Figure 7).

21. Referring to claim 13, Hammond has taught wherein the execution logic executes said unconditional jump native instruction by transferring program control to said memory address that contains the micro instruction (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

22. Referring to claim 14, Hammond has taught wherein the pipeline microprocessor comprises an x86-compatible microprocessor (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2). Hammond has not taught wherein said architectural register comprises register EAX. Hennessy has taught wherein said architectural register comprises register EAX (Hennessy page 82, paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to jump to the correct address even when the address is not known at the time the program is compiled.

23. Referring to claim 15, Hammond has taught wherein, said mode detector is configured to detect a native branch return macro instruction (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), wherein, upon detection of said native branch return macro instruction, said mode detection logic directs said translator to resume instruction translation (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

24. Referring to claim 16, Hammond has taught wherein the execution logic executes said native branch return macro instruction by transferring program control to a return memory

address (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18;, and Figure 2).

25. Referring to claim 17, Hammond has taught wherein the execution logic executes said native branch return macro instruction by transferring program control to a return memory address (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

26. Referring to claim 18, Hammond has taught a microprocessor for executing micro instructions directly from memory (Hammond column 4, lines 16-45 and Figure 1), the microprocessor comprising:

a. Translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), for receiving macro instructions from the memory (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and for decoding said macro instructions into corresponding micro instructions for execution by the microprocessor (Hammond column 17, lines 25-48 and 57-63; and Figure 7);

b. Mode detection logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), coupled to said translation logic, for detecting bypass macro instructions (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and for directing the microprocessor to execute the micro instructions directly from the memory rather than via said translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7), said bypass macro instructions comprising:

i. A native branch macro instruction, directing that program control be transferred to a target address (Hammond column 4, line 61 to column 5,

line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), wherein said translation logic decodes said native branch macro instruction into an unconditional jump native instruction directing that program control be transferred to said target address, and wherein said target address contains the micro instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2), and

ii. A native branch return macro instruction, directing that program control be transferred to a return address (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2); and

iii. An instruction router, coupled to said mode detection logic, for receiving the micro instructions (Hammond column 17, lines 25-48 and 57-63; and Figure 7), and for routing the micro instructions directly to execution logic, thereby bypassing said translation logic (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

27. Hammond has not taught wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction. Hennessy has taught wherein said memory address is explicitly prescribed by contents of an architectural register (Hennessy page 82, paragraph 2), said contents and said architectural register being prescribed by a macro instruction (Hennessy page 82,

paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to jump to the correct address even when the address is not known at the time the program is compiled.

28. Referring to claim 19, Hammond has taught wherein said mode detection logic, upon execution of said native branch macro instruction, directs said translation logic to cease decoding said macro instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

29. Referring to claim 21, Hammond has taught wherein the pipeline microprocessor comprises an x86-compatible microprocessor (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2). Hammond has not taught wherein said architectural register comprises register EAX. Hennessy has taught wherein said architectural register comprises register EAX (Hennessy page 82, paragraph 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Hennessy, would have recognized that indirect jumps allow branch, jump, and other similar types of control flow instructions to still operate correctly even when the compiler does not know the exact address to jump to at the time of compile (Hennessy page 82, paragraph 2). Therefore, a person of ordinary skill in the art at the time the invention was made would have

incorporated the indirect jumps of Hennessy in the device of Hammond in order to be able to jump to the correct address even when the address is not known at the time the program is compiled.

30. Referring to claim 22, Hammond has taught wherein said instruction router routes the micro instructions from a macro instruction bus to a micro instruction bus (Hammond column 17, lines 25-48 and 57-63; and Figure 7).

31. Referring to claim 23, Hammond has taught wherein said mode detection logic, upon execution of said native branch return macro instruction, directs said translation logic to resume decoding said macro instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

32. Referring to claim 24, Hammond has taught wherein said return address designates a next macro instruction, said next macro instruction being one of said macro instructions (Hammond column 4, line 61 to column 5, line 8; column 5, lines 54-60; column 6, lines 41-49; column 8, lines 54-60; column 18, lines 7-18; and Figure 2).

Response to Arguments

33. Applicant's arguments filed 02 September 2005 have been fully considered but they are not persuasive. Applicant argues in essence on pages 11-16

> ...Hammond does not teach any technique for completely bypassing instruction translation logic and providing programmed native instructions directly to execution logic for execution.
>
> ...But Applicant' also notes that Hennessey does not suggest, allude to, or even hint that such a technique can be employed in a microprocessor to enable a

complete bypass of translation logic thereby enabling native instructions to be provided directly to execution logic for execution.

34. This has not been found persuasive. The claims recite "directly to execution logic for execution". Execution logic, in the broadest reasonable interpretation, is logic that carries out the instructions in a computer program by a computer. A decoder is part of basic execution logic, since it control the basic operation elements by telling them what to do, e.g. it takes an instruction and produces the proper signals for the operating units to know which operation to do. The translation logic being bypassed, on the other hand, merely translates a non-native instruction into a native instruction. The native instruction still needs to pass through a decoder for the proper hardware signals to be produces for proper hardware operation, so the decoder is part of the execution logic. Also, it appears that Applicant is trying to argue that decoding the instruction after the translation logic is no longer needed. However, this is neither in the claims or the specification. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., bypassing decoding of an instruction) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Please see the accompanying information.

Conclusion

35. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

36. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

38. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

39. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
10 November 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100